Intl. J. Information Security manuscript No.
(will be inserted by the editor)

Repairing an aggregation-based smart metering system

Ricard Garra · Dominik Leibenger · Josep M. Miret · Francesc Sebé*

Received: date / Accepted: date

Abstract Smart meters inform the electricity suppliers about the consumption of their clients in short intervals. Fine-grained electricity consumption information is highly sensitive as it has been proven to permit to infer people's habits like, for instance, the time they leave or arrive home. Hence, appropriate measures have to be taken to preserve clients' privacy in smart metering systems. In this paper, we first analyze a recent proposal by Busom *et al.* (2016) and show how a corrupted substation is able to get the individual reading of any arbitrarily chosen smart meter without requiring the collaboration of any other party. After that, we propose a way to fix the mentioned security flaw which is based on adding an additional step in which the substation proves that it has properly followed all the protocol steps. Our solution is analyzed and shown to be computationally feasible for realistic parameter choices.

Keywords Encryption · Homomorphism · Privacy · Smart Metering

1 Introduction

Electricity cannot be stored in large quantities so electricity suppliers need an accurate prediction of consumption requirements and trends to properly adjust the production while avoiding electricity surplus. Fine-grained consumption data are required to that end. Smart meters are household devices that record and transmit electricity consumption readings regularly, *i.e.*, every 15 or 30 minutes. Transmission of fine-grained electricity consumption readings raises concerns about privacy, since they allow to infer behavioral patterns like the time a customer gets back home, goes to sleep, or is on vacation.

This can be achieved by means of Nonintrusive Appliance Load Monitor (NALM) [1]. By making use of signal processing methods, such techniques are able to identify the individual loads that compose a consumption trace. Hence, the information transmitted by a smart meter allows to determine, with a high degree of accuracy, the way in which appliances are used in a house.

Worldwide deployment of smart meters is a reality, and as they increase in popularity and use, appropriate solutions have to be implemented in order to provide security and privacy. An analysis and classification of privacy-preserving proposals for smart metering is given in [2].

Privacy in smart metering is a topic that is actively discussed by the industry, governments, and the research community. Any privacy-preserving solution will have to be traded off against the provisioning of advanced functionalities requiring access to personal data. That is the case for the generation of personal reports with recommendations to reduce the bill, or agreements in which the energy supplier can, for instance, temporarily pause the recharging of electric cars when the demand exceeds the supply.

(*) Corresponding author. Tel.: +34-973-702713. Fax: +34-973-702716. ORCID ID: 0000-0002-7217-5227

R. Garra · J.M. Miret · F. Sebé
Departament de Matemàtica, Universitat de Lleida, C. Jaume II, 69 E-25001 Lleida, Spain
E-mail: {garra,miret,fsebe}@matematica.udl.cat

D. Leibenger
CISPA, Saarland University, P.O. Box 15 11 50, D-66041 Saarbrücken, Germany
E-mail: dominik.leibenger@uni-saarland.de

1.1 Contribution and plan of this paper

In this paper we first describe a security flaw found in a homomorphic aggregation-based smart metering system [3]. By exploiting that flaw, a misbehaving substation is able to obtain the non-aggregated reading of any smart meter. The attack can be performed by a corrupted substation which, without being noticed by the smart meters, during a round transmits a specially manipulated message to some targeted smart meter. By combining the obtained response with the data transmitted in a forthcoming round, the substation obtains the individual reading of the targeted meter.

After describing the security flaw, we propose a way to modify [3] so that it is no longer vulnerable to the mentioned attack. Our solution is based on requiring the substation to prove, after each round, that the protocol has been properly followed. This involves the transmission of an additional message at the end of each round.

We will not discuss the problem of group selection, which is a common problem faced by all aggregation-based schemes, and is therefore out of the scope of this contribution to solve it. We are going to assume there is a minimum group size and that the substation can not manipulate at will the composition of the groups.

The remaining of the paper is organized as follows: Section 2 provides an overview on privacy-preserving technology for smart metering, with a deep focus on aggregation-based techniques. Section 3 briefly explains the original proposal by Busom *et al.* together with the security flaw we identified. We present a solution to the flaw in Section 4. In Section 5 the security of the resulting fixed proposal is formally analyzed. Section 6 presents some experimental results showing the feasibility of the fixed protocol, while Section 7 concludes the paper.

2 Related work

In a smart metering scenario, we differentiate two types of data sent from the meters to the service provider:

- Data for billing purposes,
- Data for real time consumption monitoring.

For billing purposes, identifiable communications are used for transmitting the sum of electricity consumption values (e.g., once per month) to the service provider. This is the approach taken in the smart metering architecture designed by the Germany's information security agency [4]. More elaborated solutions are required when time-of-use tariffs are used. Rial and Danezis [5] propose a privacy-preserving protocol for billing calculations. It makes use of zero-knowledge proofs (ZKP) to ensure the fee is correct without disclosing any consumption data. ZKP-based proposals like [5] and also [6] are classified as belonging to the *verifiable computation* class in [2]. All these solutions involve high computation capabilities for meters.

Regarding the transmission of fine-grained consumption readings (required for consumption monitoring), the proposals for the privacy protection of such communications can be classified into several types according to the employed technique:

- *Anonymization*: data are transmitted after removing the link between an electricity reading and the identity of the customer transmitting it [4,7,8,9]. These solutions usually require the smart meters to include the capability to manage pseudonyms and sign data. The communication overhead reduces to the additional information required for data authentication or pseudonym transmission, when required by the system.
- *Perturbation*: some random noise is added by meters to the readings before they are transmitted. The random noise will not be removed. Therefore, such solutions must be tuned to provide an adequate trade-off between accuracy and privacy [10,11]. This approach is very cheap to implement as only a simple random number generator is needed inside the meter. The communication overhead is null.
- *Aggregation*: the smart meters are partitioned into groups that aggregate their readings before they are transmitted to the company [11,12,13,14,15,16,17,18,19,20,3,21]. Data can be aggregated by making use of the homomorphic property of some cryptosystems (referred to as the *cryptographic computation* class in [2]) or by a trusted third party (belonging to the *trusted computation* class in [2]). The former solutions require the meters can perform advanced computations like public key encryption or homomorphic aggregation of encrypted data. Some of them involve very relevant computation and/or communication oveheads (See Section 2.1). The latter trusted party-based solutions do not impose any extra requirement to meters.
- *Obfuscation*: energy usage curves are altered by introducing controllable batteries and alternative generation devices within the household [22], or by the addition of devices with an adjustable consumption which inject noise to consumption traces [23]. This approach does necessarily require advanced capabilities on the smart meters, but involves the deployment of additional devices.

The proposal we cryptoanalyze and fix in this paper provides privacy by means of data aggregation using an additive homomorphic cryptosystem. Next, we review some previous proposals belonging to this class.

2.1 Aggregation-based proposals

Beyond the trivial approach of aggregating readings of a group of n smart meters using a trusted third party (see [11]), several works propose aggregation schemes based on homomorphic encryption.

García and Jacobs [12] use a combination of additive homomorphic encryption (Paillier cryptosystem) and secret sharing. When transmitting a reading, each meter $M_i, i \in \{1, \ldots, n\}$ splits its reading m_i into n shares $m_{i1}, \ldots, m_{in}$ and sends each share $m_{ij}, i \neq j$, encrypted under M_j's public key to the substation (SSt). The SSt aggregates all the shares that are encrypted under the same public key and sends each aggregate to the corresponding meter M_j. Next, each M_j decrypts the received aggregate, adds its own share m_{jj} to it, and finally sends the result back to SSt. The sum of the decrypted aggregates, now available at the SSt, equals $\sum_{i=1}^{n} m_i$. Clearly, the scheme has an elevated $O(n^2)$ communication cost per round.

A secret-sharing-based scheme proposed by Kursawe *et al.* [13] gets along with lower communication costs. In each round, p meters are deterministically chosen as *leaders*. Each meter M_i generates p random numbers $s_{i_1}, \ldots, s_{i_p}$, one share for each leader, and sends them—each encrypted with the respective leader's public key—to an aggregator (*e.g.,* SSt), which forwards them to the respective leaders. If a meter is a leader in the current round, it aggregates the $n-1$ shares it received and chooses its own so that the overall addition is 0. Each meter M_i masks its reading m_i by adding to it the sum of its shares before sending it to the aggregator. Due to the choice of the leaders' shares, the sum of all meters' masked values computed by the aggregator equals the sum of the unmasked readings. The cost per round of this proposal is $O(n \cdot p)$. For the proposal to be secure, parameter p should be large enough so that the probability of a simultaneous corruption of the chosen p leaders is negligible. Hence, a secure setting ($p \approx n$) has a negative impact on the performance.

Ács and Castelluccia [10] pursue a similar approach for aggregation, but instead of choosing global leaders, they deterministically select a number (ℓ) of pairs of smart meters in each round via a pseudo-random function. Based on pairwise keys negotiated between meters in a set-up phase, both meters of every pair compute a shared, round-specific, *dummy key* which is added by one and subtracted by the other meter to prevent decryption of non-aggregated readings. Their work is based on an encryption scheme which is additively homomorphic on both the plaintext and the key spaces [14]—a scheme which is also used in the approaches by Gómez Mármol *et al.* [15] and Vetter *et al.* [16]. Note that the readings transmitted by a meter could be determined if its ℓ chosen neighbours were all corrupted. Hence, elevating the security level has a negative impact on the system performance.

In [15], a group of smart meters are organized in a ring with a periodically changing designated smart meter acting as a *key aggregator*. Each meter chooses a random key which is forwarded to the key aggregator. The key aggregator aggregates all keys and provides the result to the energy supplier, allowing it to decrypt the homomorphically aggregated sum of encrypted readings (with the respective keys) of all the meters of the group. After each round, smart meters communicate with their neighbors in the ring to perform key renewal in such a way that the aggregated key is not changed. This solution has a large complexity as the smart meters are required to implement TLS connections, group signatures and anonymous credentials.

In [16], smart meters derive round-specific keys from a secret provided by a centralized trusted third party (which, however, is not involved in regular communication) to encrypt their readings for storage in a database operated by the energy supplier. From its knowledge about the secrets of all the meters, the trusted third party can provide *aggregation keys* to the energy supplier that allow specific aggregation operations or other database queries.

Erkin and Tsudik [17] present further approaches based on additive homomorphic encryption and secret sharing that mask readings in such a way that the readings can be extracted after aggregation. The system also allows both spatial and temporal aggregation of individual encrypted readings.

Shi *et al.* [18] utilize random values which are assigned to smart meters by a trusted dealer. They are chosen so that they sum up to zero to allow decryption only of aggregated readings. Li *et al.* [19] suggest to build a spanning tree from a collector node to individual smart meters and aggregate readings at each node of that tree using an additive homomorphic cryptosystem under the collector node's public key. In this solution, a corrupted collector is able to get individual consumptions by decrypting the messages generated by the meters before their aggregation.

In Lu *et al.* [20], the Paillier cryptosystem is used by a central entity which performs the aggregation.

The proposal by Ni *et al.* [21] distributes the n smart meters in a ring arrangement, and two rounds of com-

munication are performed to determine n polynomials containing a random noise r_i value each. The substation can then obtain the sum of all the noise values r without requiring the collaboration of any trusted third party. The readings m_i are then encrypted using a hash H of the current time slot t as $c_i = g^{m_i} H(t)^{r_i}$ and aggregated as $c = \prod_{i=1}^{n} c_i$. Schnorr signatures are employed to ensure integrity. The substation can then obtain $V = cH(t)^{-r}$, and finally get the aggregated readings from V after an easy instance of the discrete logarithm problem.

Busom *et al.* [3] propose a distributed key-generation method for an n-out-of-n threshold ElGamal. Energy consumption readings are encrypted along with some random noise which is removed during the decryption of the aggregated consumption value. This proposal is next explained in further detail in Section 3.

3 Busom *et al.* proposal

The protocol in [3] allows aggregation of the readings of a group of smart meters without relying on a trusted third party, except a certificate authority during the setup process.

It is based on the additive version of the ElGamal cryptosystem: a prime p is selected such that $p = 2q+1$, with q also prime. Then an element $g \in \mathbb{Z}_p^*$ of order q is chosen and p, q and g are the public parameters of the cryptosystem.

Given a public key $y = g^x$ ($x \in \mathbb{Z}_q^*$ is the private key), a message $m \in \mathbb{Z}_q$ is encrypted as $E_y(g^m) = (g^r, g^m \cdot y^r)$, $r \in_R \mathbb{Z}_q^*$. Decryption of such a ciphertext provides g^m as a result. A discrete logarithm is required to retrieve the cleartext m, but since the possible range for m is known and not too large, it can be done efficiently using Pollard's lambda algorithm.

The proposal in [3] employs an n-out-of-n threshold ElGamal. Each smart meter M_i stores a part x_i of the private key, and the corresponding certified public key $y_i = g^{x_i}$. Using this scheme, no trusted third party is needed to generate the group public key which is computed as $y = \prod y_i$. An ElGamal ciphertext encrypted under the group public key y can only be decrypted if all the holders of a private key fragment do collaborate.

At a given round, a smart meter M_i takes its consumption reading m_i, generates at random $r_i, z_i \in_R \mathbb{Z}_q^*$, and sends

$$E_y(g^{m_i+z_i}) = (c_i, d_i) = (g^{r_i}, g^{m_i+z_i} \cdot y^{r_i})$$

to the substation, which will aggregate all the received ciphertexts by computing

$$(c, d) = (\prod c_i, \prod d_i).$$

Then, the substation sends c to each meter M_i, which computes a partial decryption

$$T_i = c^{x_i} \cdot g^{z_i}.$$

T_i is returned to the substation, which decrypts

$$D = d \cdot (\prod T_i)^{-1} = g^m$$

with $m = \sum m_i$. Finally, the substation computes $m = \log_g D$. A sketch of the protocol can be seen in Figure 1.

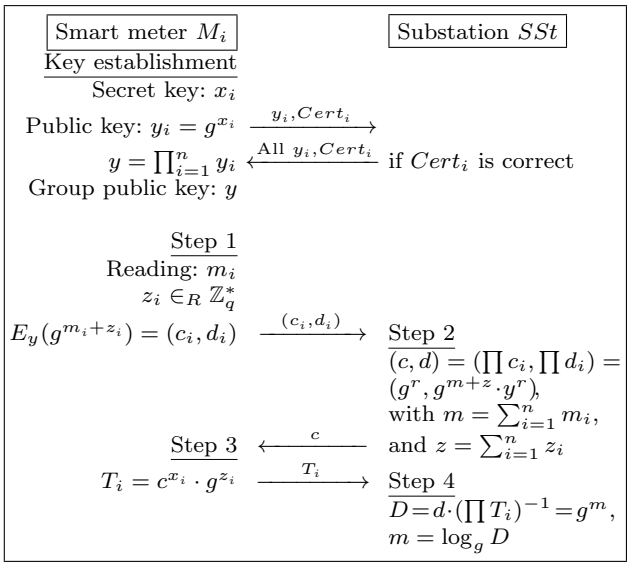


Fig. 1: Sketch of the protocol described in [3]

The security analysis in [3] considers an attacker model with a non trusted substation which may not follow the protocol steps correctly with the goal of obtaining the individual reading m_i of some targeted meter M_i, maybe with the collaboration of other corrupted smart meters. It is proven that after a proper protocol execution, the substation can only obtain the addition of (honest) meters' readings. It is also shown that if the substation sent back a value $\hat{c}$ different from c for partial decryption, it would not be able to obtain any information about $m = \sum m_i$ nor about any individual reading m_i.

However, as we will show next, these security statements are only true if the protocol is run only once. If the protocol is run for more than one round, as it is the case in smart metering, a corrupted substation can behave improperly in a given round and obtain sensitive information in subsequent rounds.

The system by Busom *et al.* [3], after fixing it, has several advantages which, in our opinion, make it one of the best proposals in the literature:

– The communication cost per round is $O(n)$ (n is the number of meters in the neighborhood). Moreover, the individual consumption reading of a meter can only be obtained after corrupting all the other meters in the neighborhood. This aspect outperforms [10,12,13], which have an $O(n^2)$ cost for an equivalent security level.
– There does not exist a trusted party dealing or storing secret keys able to compromise meters privacy if corrupted. That is the case for [16,18,19,20].
– Privacy on consumption readings is provided without requiring any underlying secure protocol (like TLS), *i.e.*, an active attacker with full access to all the communications between a substation and the meters can only get the aggregated consumption. If it was necessary to defend against an external attacker aiming to alter the received data, then some additional integrity mechanism like MACs should be included.

Regarding the capacity of meters, the deployment of such proposal would require *Group 2* meters, according to the classification given in [2].These are meters able to perform moderate operations in terms of complexity like data encryption/decryption under a public key cryptosystem, and management of certificates and keys. The solution proposed to fix the security flaw of [3] further requires some additional memory for storing a list of prime numbers (see Section 4.1).

3.1 Security flaw

We show how, in [3], a corrupted substation, without requiring the collaboration of any corrupted smart meter, can get the exact consumption reading of a targeted smart meter at a given round of the protocol. As we will see, the substation deviates from the protocol at a given round in which it does not get any information about the readings sent by the meters during that round. However, the information contained in that encrypted data can be obtained by combining it with encrypted data collected in future rounds. The security flaw and the way a corrupted substation can get advantage from it is described now.

The mentioned security flaw would permit, in a real deployment, a corrupted substation to get all the individual readings of any meter in its neighborhood. To the best of our knowledge, this security flaw had not been reported before.

At Step 2 of the protocol (see Figure 1), instead of sending c to the meters, the substation generates and sends a value $\hat{c}$ computed as $\hat{c} = g^v$ for a randomly chosen and known integer v. Each meter, upon receiving $\hat{c}$, will perform Step 3 as usual, returning $\hat{T}_i$ as a result:

$$\hat{T}_i = (\hat{c})^{x_i} \cdot g^{z_i} = (g^v)^{x_i} \cdot g^{z_i} = y_i^v \cdot g^{z_i}.$$

Since the public keys y_i are known, the substation can obtain the value g^{z_i} of any smart meter M_i at that round by computing:

$$g^{z_i} = \hat{T}_i (y_i^v)^{-1}.$$

Now, given the ElGamal ciphertext $E_y(g^{m_i+z_i}) = (c_i, d_i)$ previously sent by meter M_i, the substation is able to generate a ciphertext $E_y(g^{m_i})$ as $(c_i, d_i \cdot (g^{z_i})^{-1})$ (see Figure 3a). That ciphertext will be denoted as $E_y(g^{m_i'})$. Note that this computation could be performed simultaneously for all the meters. Nevertheless, we will now assume that the substation is only interested in the consumption reading of a targeted meter M_i. Let us refer to the components of $E_y(g^{m_i'})$ as $(c', d') = (g^{r_i'}, g^{m_i'} \cdot y^{r_i'})$.

At Step 2 of the next round, after properly aggregating all the meter ciphertexts and obtaining (c, d) as result, the substation can compute

$$(c'', d'') = (c \cdot (c')^{m_{\max}}, d \cdot (d')^{m_{\max}}),$$

with $m_{\max}$ being an upper bound on the sum of all meters' readings at any given round. Note that as $(c, d) = (g^r, g^{m+z} \cdot y^r)$, then (c'', d'') is an ElGamal ciphertext for $E_y(g^{m+z+m_i' \cdot m_{\max}})$ since

$$\begin{aligned}(c'', d'') &= (c \cdot (c')^{m_{\max}}, d \cdot (d')^{m_{\max}}) = \\ &= (g^r \cdot g^{r_i' \cdot m_{\max}}, g^{m+z} \cdot y^r \cdot g^{m_i' \cdot m_{\max}} \cdot y^{r_i' \cdot m_{\max}}) = \\ &= (g^{r+r_i' \cdot m_{\max}}, g^{m+z+m_i' \cdot m_{\max}} \cdot y^{r+r_i' \cdot m_{\max}}) = \\ &= (g^{r''}, g^{m+z+m_i' \cdot m_{\max}} \cdot y^{r''}) = E_y(g^{m+z+m_i' \cdot m_{\max}})\end{aligned}$$

for $r'' = r + r_i' \cdot m_{\max}$.

The substation can now send c'' to all the meters, and from the returned T_i'' values, it can get:

$$D'' = d'' (\prod T_i'')^{-1} = g^{m+m_i' \cdot m_{\max}}.$$

Finally, the discrete logarithm of D'' at the base g provides $m + m_i' \cdot m_{\max}$ as a result which can easily be decomposed into m and m_i' since $m < m_{\max}$. In this way, the substation gets the individual reading m_i' sent by meter M_i at the previous round in addition to the aggregated reading m of the current round.

Note that the time complexity to solve the discrete logarithm would increase by a factor of $\sqrt{m_{\max}}$ approximately. With $n \leq 128$, the problem is about 1000 times harder to solve. In our experiments, on a modern processor, such computation took less than 20 seconds.

4 Repaired proposal

In this section we propose a way to modify the protocol in [3] so that, at the end of each round, the substation is forced to provide a proof to the meters that it performed the protocol correctly. In case of failure, the meters will require a rekeying operation (Section 4.2).

The public ElGamal parameters are not modified: a prime p such that $p = 2q + 1$, with q prime, and an order-q element $g \in \mathbb{Z}_p^*$ are chosen. Additionally, in a neighborhood composed of n meters, an integer $l \geq 16$ satisfying $l \cdot n \leq \lfloor \log_2 p \rfloor - 64$ (so that $2^{l \cdot n} \ll p$) is taken.

During the setup process, the substation precomputes and stores an array $a = (a_0, \ldots, a_{m_{\max}})$ with $a_i = (g^i)^{-1}$, for $i \in \{0, \ldots, m_{\max}\}$.

At a given round, each meter M_i generates a random $z_i \in_R \mathbb{Z}_q^*$ and takes a random prime p_i whose bitlength is at most l and belongs to the subgroup of $\mathbb{Z}_p^*$ generated by g, and sends its consumption reading m_i encrypted under the group public key y as

$$E_y(p_i \cdot g^{m_i + z_i}) = (c_i, d_i) = (g^{r_i}, p_i \cdot g^{m_i + z_i} \cdot y^{r_i})$$

to the SSt, which aggregates all the received ciphertexts into $(c, d) = (\prod c_i, \prod d_i)$.

Steps 2 and 3 of the original proposal remain unmodified. The SSt sends c to each meter and receives all the $T_i = c^{x_i} \cdot g^{z_i}$ per response. Then, it computes $D = d \cdot (\prod T_i)^{-1} = P \cdot g^m$, with $P = \prod p_i$ and $m = \sum m_i$. Now the substation just needs to find the probably unique m such that $D \cdot a_m < 2^{l \cdot n}$, obtaining then the values m and $P = D \cdot a_m$.

The substation factorizes P and stores its prime factors in an ordered list p_v, which is then sent to all the meters. Each meter M_i will then check that the prime p_i it chose is in p_v and $|p_v| = n$. If the check fails, the meter requests an execution of a key renewal operation (Section 4.2). This serves as an assurance that the substation has performed the protocol correctly, as the list p_v of primes can not be obtained by other means.

Note that, as a result of a key renewal operation, the current group public key $y = g^x$ is replaced with a new one $\bar{y} = g^{\bar{x}}$, so that any ciphertext $E_y(g^{m_i'})$ becomes the encryption of an undetermined cleartext under the new public key. This is so because

$$E_y(g^{m_i'}) = (g^{r_i'}, g^{m_i'} \cdot y^{r_i'}) = (g^{r_i'}, g^{m_i'} \cdot g^{x \cdot r_i'} \cdot g^{-\bar{x} \cdot r_i'} \cdot g^{\bar{x} \cdot r_i'}) =$$

$$= (g^{r_i'}, g^{m_i' + r_i'(x - \bar{x})} \cdot \bar{y}^{r_i'}) = E_{\bar{y}}(g^{m_i' + r_i'(x - \bar{x})}),$$

being the difference between private keys $(x - \bar{x})$ a random value unknown to any party. Hence, after rekeying, any ciphertext collected in previous rounds becomes the encryption of a valueless random message.

The transmission of p_v to all the meters was not in the (weak) original proposal [3]. Since p_v is storing the prime factors of P, which is an element of $\mathbb{Z}_p$, its bitlength is at most that of an element of $\mathbb{Z}_p$. Hence, the communication overhead introduced is $O(n)$ if the transmission from the substation to the meters is unicast, or $O(1)$ if a broadcast channel is available. In any case, the communication complexity of the proposal remains the same, $O(n)$.

Next, we provide a more detailed description of the repaired proposal.

4.1 Setup

- *System parameters:* An ElGamal cryptosystem is set up by choosing two large primes p and q, with $p = 2q + 1$, and an order q element $g \in \mathbb{Z}_p^*$. An integer $l \geq 16$ satisfying $l \cdot n \leq \lfloor \log_2 p \rfloor - 64$ is also chosen.
- *Digital certificates:* each meter M_i stores a certificate CertCA_i with a private/public key pair whose public key is certified by a trusted authority. Such key pair will allow the meter to self-sign its public keys.
- *Prime table precomputation:* each meter M_i stores a table with all the prime integers whose bitlength is at most l and belong to the subgroup of $\mathbb{Z}_p^*$ generated by g.
- *Array precomputation:* The substation computes and stores an array $a = (a_0, \ldots, a_{m_{\max}})$ with

$$a_i = (g^i)^{-1}, \text{for } i \in \{0, \ldots, m_{\max}\},$$

where $m_{\max}$ is an upper bound on the aggregated consumption of all meters' readings at a given round.

4.2 Key establishment

- Each meter M_i takes a random $x_i \in \mathbb{Z}_q^*$ as its current private key, and computes the corresponding public key $y_i = g^{x_i}$. A certificate Cert_i is issued for y_i using the key pair in CertCA_i. It then sends the key establishment data $\text{KED}_i = (y_i, \text{Cert}_i, \text{CertCA}_i, \text{PK}(x_i))$ to the substation, with $\text{PK}(x_i)$ being a proof of knowledge on secret key x_i (a signed timestamp would serve to that end).
- For each meter, the substation verifies the certificate CertCA_i under the public key of the certificate authority, then it verifies Cert_i under the public key in CertCA_i, and it finally checks the proof of knowledge on the private key related to the public key y_i. If all the checks are found correct it forwards all

KED_i to all the meters, which will perform the same checks.
– Finally, the substation and all the smart meters compute the group public key:

$$y = \prod_{i=1}^{n} y_i.$$

4.3 Electricity consumption transmission

Every time period (e.g., every 15 or 30 min) the SSt sends a message to all the smart meters requesting their electricity measurements. Let m_i denote the reading of smart meter M_i at the current time period.

1. Each M_i generates a random $z_i \in_R \mathbb{Z}_q^*$ and takes a random prime $p_i < 2^l$ from its internal memory. It then composes and sends, to the SSt, the ciphertext

$$E_y(p_i \cdot g^{m_i+z_i}) = (c_i, d_i) = (g^{r_i}, p_i \cdot g^{m_i+z_i} \cdot y^{r_i}).$$

2. The SSt aggregates all the received messages as

$$(c, d) = (\prod c_i, \prod d_i) = (g^r, P \cdot g^{m+z} \cdot y^r),$$

with $m = \sum m_i$, $z = \sum z_i$ and $P = \prod p_i$, and sends c to each M_i.
3. Each M_i computes $T_i = c^{x_i} \cdot g^{z_i}$ and sends the result to the SSt. At this point, each M_i removes z_i from its memory.
4. The SSt computes

$$D = d \cdot (\prod T_i)^{-1} = P \cdot g^m.$$

Since $P < 2^{l \cdot n}$, the SSt determines the received m as the probably only one satisfying

$$D \cdot a_m < 2^{l \cdot n}.$$

Since $D \cdot a_m = P$, the SSt can now factor P and send its prime factors in an ordered list p_v to each M_i.
5. Each M_i checks that $p_i \in p_v$ and $|p_v| = n$. If some meter's check fails, it requests the execution of the key establishment protocol (see Section 4.2) and all the meters' public keys and the group public key are renewed.

In Figure 2 a sketch of the repaired protocol can be seen, with $\mathbb{P}$ denoting the internal table storing prime numbers.

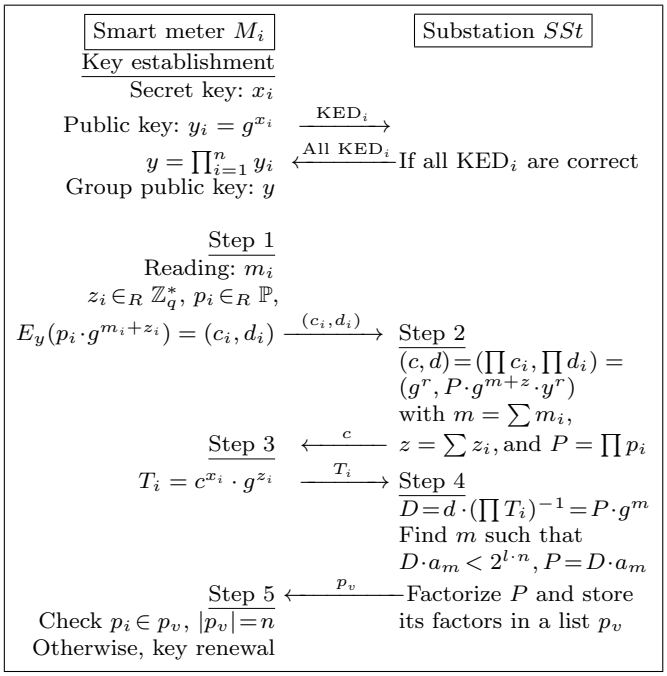


Fig. 2: Sketch of the repaired protocol

5 Security analysis and parameter tuning

The repaired proposal inherits most of its security properties from the original proposal [3], like the impossibility for a corrupted substation to get any information about individual consumption readings after just one round of the protocol. That is, after just one round, a corrupted substation together with a subset of corrupted smart meters, can not learn more than the sum of the readings of the rest of honest meters.

However, as detailed in Section 3.1, if the original protocol is executed more than one round, the individual readings of individual smart meters can be obtained. Next, we provide a security analysis of our repaired proposal.

In the repaired proposal, a meter M_i chooses a small prime p_i that is part of the ciphertext it transmits to the substation. Since the small primes employed in the repaired proposal belong to the subgroup of $\mathbb{Z}_p$ generated by g, then it holds that $p_i = g^{v_i}$ for some value v_i. Then, the ciphertext $E_y(p_i \cdot g^{m_i+z_i})$ can be expressed as $E_y(g^{v_i} \cdot g^{m_i+z_i})$ which is equal to $E_y(g^{m'_i+z_i})$ with $m'_i = v_i + m_i$. Hence, the ciphertexts transmitted in the repaired proposal are of the same form as in the original one.

Therefore, we can apply the Proposition 4 in [3], which is next reproduced.

Proposition 1 *Let us consider a neighborhood composed of n meters so that a subset of them, $M_1, \ldots, M_{n'}$,*

$n' \leq n$, acts honestly. Obtaining a partial aggregation of their consumption values is as hard as solving a CDH problem.

In our case, the mentioned proposition implies that if an attacker has corrupted meters $M_{n'+1}, \ldots, M_n$, the only information it can obtain about the honest ones $(M_1, \ldots, M_{n'})$ is $g^{\sum_{i=1}^{n'} m'_i} = \prod_{i=1}^{n'} p_i \cdot g^{\sum_{i=1}^{n'} m_i}$.

After a proper protocol execution, the substation obtains $\prod_{i=1}^{n} p_i \cdot g^{\sum_{i=1}^{n} m_i}$ and the previous partial message can be computed if the corrupter meters tell their p_i and m_i values. Hence, the protocol guarantees that, after a protocol run, the attacker can not get any information about the cleartexts but that given by a proper execution.

In the repaired proposal, at the end of a protocol round, the substation is required to prove the knowledge of the prime numbers chosen by the meters. Otherwise, a rekeying procedure will be run. From the previous analysis, a corrupted substation can only get the primes p_i after a proper protocol execution, otherwise, it gets no information about them and the only remaining option is to guess them at random. Figure 3b depicts how the attack described in Section 3.1 fails when a targeted smart meter notices that its prime p_i is not in the received list.

Considering that each smart meter M_i chooses p_i independently of each other, and assuming that only a subset of $n' < n$ meters are honest (so that the remaining $n - n'$ primes are known to the substation), the substation should return a list containing the n' unknown ones. The probability that such an n-elements list contains the n' unknown ones is $\left(\frac{n}{k}\right)^{n'}$ with k being the amount of primes smaller than 2^l that belong to the subgroup of $\mathbb{Z}_p$ generated by g. Note that $k \approx \pi(2^l)/2$ with $\pi(x)$ being the prime-counting function. Since half of the elements of $\mathbb{Z}_p$ belong to the subgroup of g then, approximately half of the primes smaller than 2^l belong to that subgroup.

The value for l should be set to avoid a too easy guessing of such primes, so that we recommend $l \geq 16$. Also, l should satisfy $2^{l \cdot n} \ll p$ so that the probability that there exist more than one a_m satisfying $D \cdot a_m < 2^{l \cdot n}$ is negligible and it is possible to detect the correct $P = \prod p_i$ from its length. To that end, we recommend to take

$$l \leq \frac{\lfloor \log_2 p \rfloor - 64}{n}.$$

In this way, the probability that a random element in $\mathbb{Z}_p$ is smaller than $2^{l \cdot n}$ is less than 2^{-64}. In a setting with $m_{\max} = 2^{20}$, the expected amount of protocol executions to be performed before the first failure would be around 2^{44}. Assuming a transmission each 15 minutes, the expected time before the first failure would be larger than $5 \cdot 10^8$ years. The only consequence of such failure would be the execution of an unnecessary rekeying operation.

If we take the smallest recommended value for l, *i.e.* $l = 16$, we get $k \approx \pi(2^{16})/2 = 3271$ (the exact amount depends on the choice of p). All such primes could be stored inside each meter as 16 bits integers, requiring less than 7 KiB of memory, making it easy for the meters to take any of them at random. In a neighborhood composed of $n = 128$ meters, if $n' = 10$ meters are honest, the probability to guess all their primes is around $8.4 \cdot 10^{-15}$, and even with $n' = 5$, the probability is still less than $9.2 \cdot 10^{-8}$.

6 Experimental analysis

To verify the feasibility of our protocol in practice, we performed experiments on a computer with an Intel i5 4460 (3.4 GHz turbo frequency, 4 cores) processor, using the GMP library (The GNU Multiple Precision Arithmetic Library) and OpenMP in a C++ program.

The time to generate the array a with $m_{\max} = 2^{20}$, which is enough for up to 128 smart meters, using a 2048 bits prime p, was 20 seconds. This was stored in a 260 MiB file. If we take a 3072 bits prime, we can potentially accommodate up to 180 smart meters, and the corresponding array a requires 550 MiB of storage. The array was generated in 60 seconds.

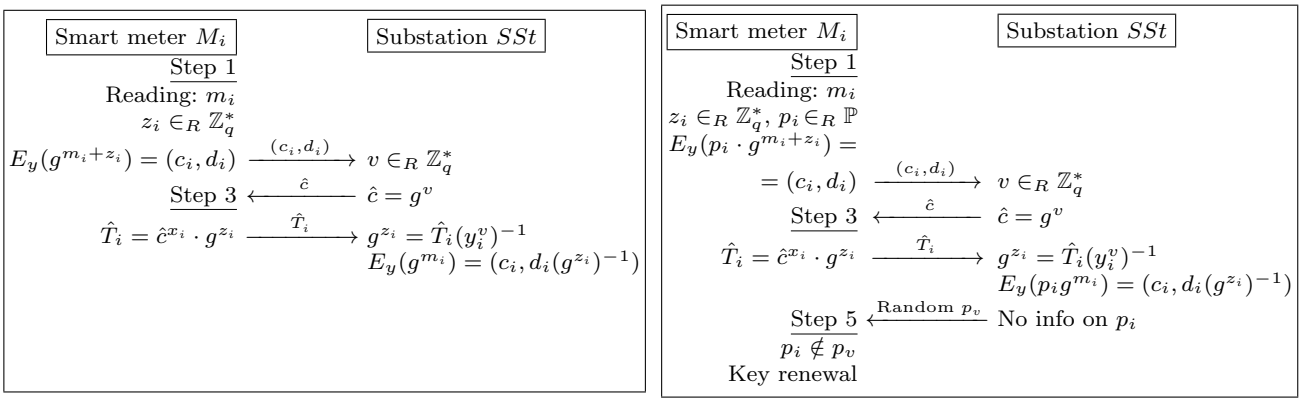
The other time consuming task is that in Step 4, in which the substation multiplies D by the elements in array a until getting a result whose length in bits is smaller than $l \cdot n$. As we can see in Figure 4, the maximum time required to find a_m scales linearly with the number of smart meters n, for a fixed prime p. The size of l does not affect this time.

With $l \leq \frac{\lfloor \log_2 p \rfloor - 64}{n}$, and by choosing p to be 2048 bits long, a neighborhood can include up to 124 smart meters. If more meters are to be accommodated, a larger p must be chosen.

By looking at Figure 4 we conclude that a reasonable value for p is a 2048 bit prime. This setting provides a high enough security while keeping the computing time at around half a second. For more meters, it could be advisable to split the community into several groups rather than increasing p.

7 Conclusions

In this paper, we describe a vulnerability found in a homomorphic aggregation-based proposal [3] for reporting the consumption of a group of smart meters. The



(a) First round of a successful attack on the original protocol

(b) Failed attack against the repaired protocol

Fig. 3: First round of an attack against the original (left) and the repaired (right) proposals

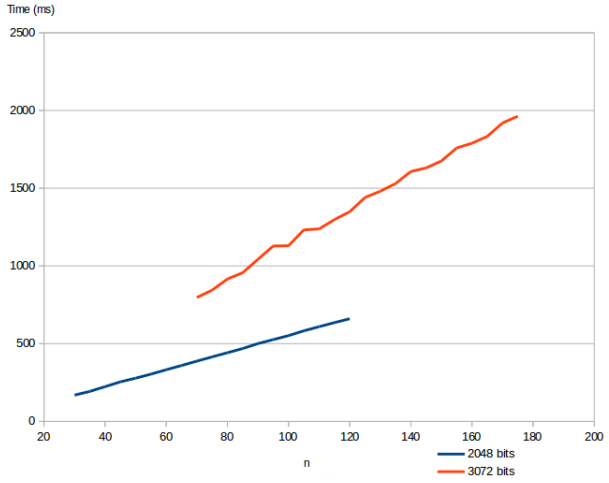


Fig. 4: Maximum time to perform Step 4 and find m and P depending on the number of meters n and the size in bits of p.

security flaw allowed a corrupted substation to obtain individual readings of meters.

After that, we propose a way to slightly modify the original proposal so as to fix the mentioned security flaw. The modified proposal is very similar to the original one. It adds an additional transmission at the end of each round in which the substation proves that it has followed the protocol properly. The repaired proposal is proven to maintain the privacy of smart meters throughout several execution rounds.

The computations carried out by the meters are basically the same than in the original proposal, but now they have to choose a small random prime, which can be taken from a small precomputed table. The substation has a higher computation overhead, specially during the setup in which it has to precompute and store a relatively large array. Nevertheless, that computation is performed only once at setup time.

Compliance with ethical standards

Funding: This study was funded by the European Regional Development Fund of the European Union in the scope of the "Programa Operatiu FEDER de Catalunya 2014–2020" (project number COMRDI16-1-0060), by the Spanish Ministry of Science, Innovation and Universities (project number MTM2017-83271-R), and by the Federal Ministry for Economic Affairs and Energy of Germany in the SINTEG project DESIGNETZ (project number 03SIN224).

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Acknowledgements This study was funded by the European Regional Development Fund of the European Union in the scope of the "Programa Operatiu FEDER de Catalunya 2014–2020" (project number COMRDI16-1-0060), by the Spanish Ministry of Science, Innovation and Universities (project number MTM2017-83271-R), and by the Federal Ministry for Economic Affairs and Energy of Germany in the SINTEG project DESIGNETZ (project number 03SIN224).

References

1. Hart, G.W.: Nonintrusive appliance load monitoring, Proc. of the IEEE 80, 1870–1891 (1992).
2. Rubio, J.E., Alcaraz, C., López, J.: Recommender system for privacy-preserving solutions in smart metering, Pervasive and Mobile Computing 41, 205-218 (2017). doi: 10.1016/j.pmcj.2017.03.008.

3. Busom, N., Petrlic, R., Sebé, F., Sorge, C., Valls, M.: Efficient smart metering based on homomorphic encryption, Computer Communications 82, 95-101 (2016). doi:10.1016/j.comcom.2015.08.016.
4. Stegelmann, M., Kesdogan, D.: GridPriv: a smart metering architecture offering k-anonymity, 11th Trust, Security and Privacy in Computing and Communications, "TrustCom'12", pp. 419–426 (2012). doi:10.1109/TrustCom.2012.170.
5. Rial, A., Danezis, G.: Privacy-preserving smart metering, Technical Report MSR-TR-2010-150, Microsoft Research (2010).
6. Molina-Markham, A., Shenoi, P., Fu, K., Cecchet, E., Irwin, D.: Private memoirs of a smart meter, 2nd ACM W. on Embedded Sensing Systems for Energy-Efficiency in Building, pp. 61–66 (2010).
7. Efthymiou, C., Kalogridis, G.: Smart grid privacy via anonymization of smart metering data, Smart Grid Communications (SmartGridComm), 2010 First IEEE Intl. Conf. on, pp. 238–243 (2010). doi:10.1109/SMARTGRID.2010.5622050.
8. Finster, S., Baumgart, I.: Pseudonymous smart metering without a trusted third party, 12th Trust, Security and Privacy in Computing and Communications, "TrustCom'13", pp. 1723–1728 (2013). doi:10.1109/TrustCom.2013.234.
9. Petrlic, R.: A privacy-preserving concept for smart grids, Sicherheit in vernetzten Systemen 18, B1–B14 (2010).
10. Ács, G., Castelluccia, C.: I have a dream! (differentially private smart metering), Information Hiding (LNCS), vol. 6958, Springer-Verlag, Berlin Heidelberg, pp. 118-132 (2011). doi:10.1007/978-3-642-24178-9_9.
11. Bohli, J.-M., Sorge, C., Ugus, O.: A privacy model for smart metering, Proc. of the First IEEE Intl. W. on Smart Grid Communications (in conjunction with IEEE ICC) (2010).
12. García, F., Jacobs, B.: Privacy-friendly energy-metering via homomorphic encryption, Proc. of 6th Intl. Conf. on Security and Trust Management (LNCS), vol. 6710, Springer-Verlag, Berlin Heidelberg, pp. 226–238 (2011) . doi:10.1007/978-3-642-22444-7_15.
13. Kursawe, K., Danezis, G., Kohlweiss, M.: Privacy-friendly aggregation for the smart-grid, Proc. of Privacy Enhancing Technologies (LNCS), vol. 6794, Springer-Verlag, Berlin Heidelberg, pp. 175–191 (2011). doi:10.1007/978-3-642-22263-4_10.
14. Castelluccia, C., Mykletun, E., Tsudik, G.: Efficient aggregation of encrypted data in wireless sensor networks, Proc. of The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 109–117 (2005). doi:10.1145/1525856.1525858.
15. Gómez Mármol, F., Sorge, C., Petrlic, R., Ugus, O., Westhoff, D., Martínez Pérez, G.: Privacy-enhanced architecture for smart metering, Intl. J. of Information Security 12:2, 67–82 (2013). doi:10.1007/s10207-012-0181-6.
16. Vetter, B., Ugus, O., Westhoff, D., Sorge, C.: Homomorphic Primitives for a Privacy-Friendly Smart Metering Architecture, Proc. of the Intl. Conf. on Security and Cryptography, pp. 102–112 (2012).
17. Erkin, Z., Tsudik, G.: Private Computation of Spatial and Temporal Power Consumption with Smart Meters, Proc. of Applied Cryptography and Network Security (LNCS) , vol. 7341, Springer-Verlag, Berlin Heidelberg, pp. 561–577 (2012).
18. Shi, E., Chow, R., Chan, T-H.H., Song, D. , Rieffel, E.: Privacy-preserving aggregation of time-series data, Proc. of Network and Distributed System Security Symposium. The Internet Society (2011).
19. Li, F., Luo, B., Liu, P.: Secure information aggregation for smart grids using homomorphic encryption, First IEEE Intl. Conf. on Smart Grid Communications, pp. 327–332 (2010). doi:10.1109/SMARTGRID.2010.5622064.
20. Lu, R., Liang, X., Li, X., Lin, X., Shen, X.: An efficient and privacy-preserving aggregation scheme for secure smart grid communications, IEEE Trans. on Parallel and Distributed Systems, 23:9, 1621–1631 (2012). doi:10.1109/TPDS.2012.86.
21. Ni, J., Zhang, K., Lin, X., Shen, X.: EDAT: Efficient Data Aggregation without TTP for Privacy-Assured Smart Metering, IEEE Intl. Conf. on Communications (2016). doi:10.1109/ICC.2016.7510611.
22. Kalogridis, G., Efthymiou, C., Denic, S.Z., Lewis, T.A., Cepeda, R.: Privacy for smart meters: Towards undetectable appliance load signatures, Smart Grid Communications (SmartGridComm), 2010 First IEEE Intl. Conf. on, pp. 232–237 (2010).
23. Egarter, D., Prokop, C., Elmenreich, W.: Load hiding of household's power demand, IEEE Intl. Conf. on Smart Grid Communications, pp. 854–859 (2014).